Amendments to the Specification:

Page 1, please replace the Title with the following amended Title:

~~SPECULATIVE LOAD INSTRUCTIONS WITH RETRY~~
MICROPROCESSOR FOR EXECUTING SPECULATIVE LOAD INSTRUCTIONS WITH RETRY OF SPECULATIVE LOAD INSTRUCTION

Pages 2-3, please replace the third paragraph, beginning at line 15 and extending to page 3, line 6, with the following amended paragraph:

In typical computer systems, these limitations severely impede the ability to do loads early enough to get significant increases in speed. The ~~Intel~~ INTEL IA-64 architecture uses an instruction called a speculative load (ld.s) that is executed by the processor early in the instruction stream. Each general purpose register has an associated bit known as the Not a Thing (NaT) bit (NaTVal for floating point registers) that keeps track of whether or not the data is valid. Later in the instruction stream, when the load is known to be necessary, a speculation check instruction (chk.s) is executed which checks the NaT bit to confirm that the data is still valid (NaT = 0). However, if invalid data is detected (NaT = 1) recovery operations are commenced. The NaT bit is set when the load fails (e.g. incomplete or incorrect data). This bit is propagated with the load instructions through the pipeline. Typically, when the processor is ready to use the loaded data it checks the NaT bit and if it is set (equals logical 1), then recovery code is called and the data value must be recomputed. That is, the speculative check instruction (chk.s) tests for a deferred exception token (NaT = 1). If none is found, then the speculative calculation was successful, and execution continues normally. However, if the NaT bit is set then the speculative calculation was unsuccessful and must be redone. In this case, the chk.s instruction branches to a new address to invoke recovery code (program that contains a copy of the speculative calculation with non-speculative loads). In this manner, it can be seen that the

existing chk.s instruction in the IA-64 instruction set will always cause the branch to recovery to be taken when the deferred exception token is found (NaT = 1).

Pages 8-9, please replace the second paragraph, beginning at line 21 and extending to page 9, line 23, with the following amended paragraph:

Referring to Figure 1, a typical data processing system is shown which may be used in conjunction with the present invention. A central processing unit (CPU) 10 may include an ~~Intel~~ INTEL Itanium microprocessor that executes the IA-64 instruction set and is commercially available from Intel Corporation, as well as any other commercially available microprocessor that performs speculative processing and would benefit from the ability to check such speculative operation, such as the POWER 3 microprocessor from International Business Machines, Corp. The CPU 10 is shown interconnected to the various other system components by a system bus 12. Read only memory (ROM) 16 is connected to CPU 10 via bus 12 and includes the basic input/output system (BIOS) that controls the basic computer functions. Random access memory (RAM) 14, input/output (I/O) adapter 18 and communication adapter 34 are also connected to system bus 12. I/O adapter 18 may include a small computer system interface (SCSI) adapter that communicates with a disk storage device 20. Communications adapter 34 may be a network card that interconnects bus 12 with an outside network. Adapter 34 may also include an I/O port that allows a connection to be made, through a modem 40, or the like to enable the data processing system to communicate with other such systems via the Internet, or other communications network (LAN, WAN). User input/output (I/O) devices are also connected to system bus 12 via user interface adapter 22 and display adapter 36. Keyboard 24, track ball 32, mouse 26 and speaker 28 are all interconnected to bus 12 via user interface adapter 22. Display monitor 38 is connected to system bus 12 by display adapter 36. In this manner, a user is capable of inputting to the system through keyboard 24, trackball 32 or mouse 26 and receiving output from the system via speaker 28 and display 38. Additionally, an application program 37 is running on operating system (OS) 39, such as the AIX, DOS, OS/2, ~~Windows~~ WINDOWS operating system, or the like. Operating system 39 is shown running on CPU 10 and used to

coordinate the functions of the various components illustrated by Figure 1. Program 37 is used to provide specific functions desired by the user, such as word processing, spread sheet, calendar, or the like. Those skilled in the are will understand that program 37 and OS 39 are running on CPU 10 in a binary format, as output by a compiler 40, which transforms the source code from a programmer into a form which is executable by a microprocessor.

Page 10, please replace the first paragraph with the following amended paragraph:

The following description is a high level summary of the operation of an ~~Itanium~~ ITANIUM microprocessor, which is used herein as merely one example of a type of processor capable of benefiting from the present invention. Of course, any data processing system which may use the speculative check function of the present invention is contemplated by the scope of the appended claims. For additional information regarding the ~~Intel Itanium~~ ITANIUM microprocessor, please refer to the *Intel Itanium Architecture Software Developer's Manual, vols 1-4, July 2000; Itanium Processor Microarchitecture Reference, August 2000,* and the *Intel Itanium Processor Hardware Developer's Manual, May 2001*, all available from the Intel Corporation.

Pages 13-14, please replace the third paragraph, beginning at line 27 and extending to page 13, line 5, with the following amended paragraph:

Referring to Figure 5, the speculative advanced load hardware for the IA-64 ~~Itanium~~ ITANIUM processor is shown. The virtual address for the operand data corresponding to an advanced load instruction is input to translation lookaside buffer (TLB) 84, where the virtual address is translated to a physical address and entered into ALAT 85. As described previously, the NaT bit propagates with the advanced load and the speculative load status instruction to exception logic 86 which determines if the data from the advance load is valid. The speculation check instruction (chk.s) then calls a recovery routine if the NaT bit is set, indicating the data is invalid.